**Covid – US**

This repository includes the data collected of the global pandemic started in 2019 Coronavirus. **Coronavirus** is a family of viruses that can cause illness, which can vary from common cold and cough to sometimes more severe disease and can lead to death.

This problem statement is based on the cases of the country United States of America only. As of now there are more than 510K deaths caused by Coronavirus only in US.

**Problem Definition**

**Our goal or objective** was to find the number of total deaths and recovered people based on the other features information given to us, which includes Province\_State, Country\_Region , Last\_Update, Lat, Long\_, Confirmed, Deaths, Recovered, Active, FIPS, Incident\_Rate, People\_Tested, People\_Hospitalized, Mortality\_Rate, UID, Testing\_Rate, Hospitalization\_Rate.

**Data Analysis and EDA Conclusions**

When we started analysis of Dataset given to us we found that out of our 17 independent features from total of 18 features 2 of them were categorical features which were “Province\_State” and “ISO3A”, which were needed to be treated, apart from that there were many features consisting of missing values in the dataset which are also needed to be treated.

I started the EDA part of the dataset by initially treating the missing values, for which I first try to figure out which feature has what percentage of missing values with respect to our dataset. (as there are only 58 rows or observations in our dataset i.e. for the respective states of America and few other neighbouring country so it was not feasible to drop any observation.) After figuring out the same I found that there were 2 features which have approx. half of their information missed which were “People\_Hospitalized”, “Hospitalization\_Rate”. Also there were 2 features which have same observation in the entire dataset so this won’t make any impact on the prediction so I decided to drop the features “Last\_Update” and “Country\_Region” now there were 16 features in total.

I also had to treat the categorical features which were “Province\_State” and “ISO3A” so that they can be accepted as input parameters by the Machine Learning Model which will be using later on, for the treatment of these categorical features I used One hot Encoding technique for which I used Panda’s “get\_dummies” function.

The next step of the Exploratory Data Analysis was to handle the missing values, but before that I also had to check the skewness present in the dataset for which I simply used Seaborn library’s function “distplot” function which gave me the visual representation of the skewness present in the features of dataset, after plotting the skewness It was visible that the data which was given to us had many outliers which leads to skewness in dataset, which was also needed to be treated as otherwise it will leads to wrong predictions.

As there were very less observations so I couldn’t drop any observations also I found out that there were no such groups present on which our dataset could be categorised so for the treatment of missing values I tried to replace the missing values of each feature with the mean of those respective features. On a special case we couldn’t fill the missing values of Lat and Long features with the mean of those features so we tried to keep them blank.

After completing this step we have treated all our features i.e. our categorical features have been encoded using One Hot Encoding Technique and the new encoded features were created with different names in separate dataframe with names “**Province\_State\_n**” and “**ISO3\_n**” based on both of the encoded features and our features with missing values have been replaced with the mean values of those features and there were no Nan values left in the dataset. Now I have to merge all the required data into one dataframe i.e. our encoded features with the original dataframe, for which I used simple Panda’s method Concat. After concatenating the dataframes we also had to drop the unwanted features i.e. original categorical features “**Province\_State**” and “**ISO3**” as in place of them we are using encoded columns, after merging the dataset there was cast increment in the features of our dataset our dataset with 16 features have been upgraded to 76 features out of which 75 will be used as independent features for prediction.

Now our dataset was prepared which has no missing values and no categorical columns but the features were present with skewness which still needs to be treated. But before that we had to segregate our independent and dependent features so that there won’t be any leakage in the data while building model and our predictions will be correct.

**First we tried to find out the total Deaths** so we used variable “x” to represent our independent features and variable “y” to represent our dependent variable (i.e. “Deaths”). For the treatment of skewness we import power\_transform library of sklearn.preprocessing and used “yeo-johnson” method for the handling the skewness.

Now our dataset (train and test dataset) our ready to be used for predictions. So I split the train and test dataset in the ration of 80% and 20% respectively. I also tried to find the best random state for the prediction so I use Linear regression for initial prediction (As this is a regression problem ) and tried to compare the “**r2\_score**” of the train and predicted train with the “**r2\_score**” of the test and predicted test up to 1000 times, but sadly there was no match so I defined random state by my own.

After splitting our dataset I started applying multiple algorithms started with Linear Regression along with the regularisation technique Lasso Regression from library “sklearn.linear\_model” module “Lasso” for better results I tried to optimise the parameters of the algorithm (alpha, random\_state) by using Grid Search CV and the using the best parameters for the dataset generated by Grid Search CV after fitting the train and test data with the new better suited parameters we received R2 score of 30% which was very less and was not acceptable so our Lasso regression model was rejected.

Later we used the Decision tree algorithm of package “sklearn.tree” and repeat the steps starting with parameter optimisation (“criterion”, “splitter”) using Grid Search CV, when I received the best parameters of the decision tree technique for the given algorithm I fit the model again with this technique and received R2 score of 40% which was better than previous approach but still wasn’t acceptable as it was still very low.

We tried to improve our model using ensemble techniques for which I used the algorithm of Random Forest which is basically based on Decision tree itself, from package “sklearn.ensemble” module “RandomForestRegressor” and then again repeat the steps starting with parameter optimisation (“criterion”, “n\_estimators”) using Grid Search CV, when I received the best parameters of the Random Forest technique for the given algorithm I fit the model again with this technique and received R2 score of 86% which was far better than both of the previous approach and can be accepted for model building, so we used Random Forest model for the prediction of Deaths.

**Now we have to predict the Recovered cases** using the same data so we segregated the data set again into x and y but this time we used “Recovered” as our independent feature, and repeated the steps of fitting the different models on our x and y data and received different R2 scores but again we found the best score using Random Forest Algorithm, So we again used Random Forest model for the prediction of Recovered.

At last we saved both the models for Death prediction and Recovered prediction using Random Forest algorithm in form of a pickle file using package “pickle”.